



ADVANCED PROGRAMMING INFORMATION



18644 Oxnard Street, Tarzana, CA 91356 • (213) 996-5060

800155/882



INTRODUCTION TO API

This manual serves as a guide to the advanced programmer. It describes RAM memory locations and their use as applies to the Apple-Cat II. Also included are some routines showing use of these memory locations, or registers as they're called, to control the modem. With this information, the advanced programmer can write or adapt machine language or BASIC routines to accomplish the individual tasks that he or she may desire.

The manual is organized by the registers, in ascending order, followed by the routines using them, collectively called 'CATPACK.' Each of the registers is broken down into its bit components and the function of each is described.

There are two indexes included in this appendix—one ordered numerically by register, the other ordered alphabetically by bit component description. Any register or bit component can be located easily. For example, if the register name or address is known, information in the first index will indicate where it can be found; if a particular bit component is needed, the second index will indicate its whereabouts.

'CATPACK' is a package of routines that shows how to control the Apple-Cat II through machine language. These routines show the use of the registers described in the previous section. One may want to modify them to meet specific applications requirements.



REGISTER INDEX

Base Address*

Hex	Decimal	Read	Write	Page
APPLE-CAT II				
\$C080	-16256	SWBYT	DACBYT	4,5
\$C081	-16255	TONBYT	SQUBYT	5,7
\$C082	-16254	—	SHBYT	8
\$C083	-16253	—	BSRBYT	9
\$C089	-16247	—	RECBYT	10
\$C08A	-16246	—	CON	11
\$C08B	-16245	INDAT	SPDBYT	12
\$C08C	-16244	—	COM	13
\$C08D	-16243	ACBYT	XMTBYT	14,15
\$C08E	-16242	—	OUTDAT	15
\$C08F	-16241	STATUS	DEVBYT	16,17

212 CARD

\$C080	-16256	—	212BYT	18
--------	--------	---	--------	----

*Add \$N0 to hex base address or 16*N to decimal base address (where N = slot of Apple-Cat II or 212 card) to obtain slot-adjusted address.

BIT COMPONENT INDEX

Bit Component	Register	Page
Audio detect	SWBYT	4
BSR 60 Hz input	SWBYT	4
BSR 120 KHz output	BSRBYT	9
Carrier detect	SWBYT	4
Carrier detect timing	RECBYT	10
Clear DV	SHBYT	8
Clear to send	ACBYT	14
DAC bits	DACBYT	5
DV from TouchTone	SWBYT	4
Firmware switches	SWBYT	4
From UART txd to	DEVBYT	17
IRQ ring enable	XMTBYT	15
IRQ 29.12 Hz enable	BSRBYT	9
IRQ 29.12 Hz status	TONBYT	5
Microphone squelch	SQUBYT	7



Continued

Bit Component	Register	Page
Mode control bits	RECBYT	10
Receive baud rate	SPDBYT	12
Receive data from UART	INDAT	12
Ring detect	ACBYT	14
Speaker squelch	SQUBYT	7
Switch hook	SHBYT	8
Tape recorder motor control	SQUBYT	7
TouchTone bits	TONBYT	5
Transmit baud rate	SPDBYT	12
Transmit data to UART	OUTDAT	15
Transmitter mode	XMTBYT	15
UART command—receiver	COM	13
UART command—transmitter	COM	11
UART control—parity enable	CON	11
UART control—parity	CON	11
UART control—# stop bits	CON	11
UART control—# data bits	CON	11
UART rxd to	DEVBYT	17
UART status—overrun error	STATUS	16
UART status—framing error	STATUS	16
UART status—parity error	STATUS	16
UART status—tx data register empty	STATUS	16
UART status—rx data register full	STATUS	16
UART status—transmit IRQ	STATUS	16
UART status—receive IRQ	STATUS	16
UART status—ring IRQ	STATUS	16
212 analog loopback	212BYT	18
212 carrier detect	ACBYT	14
212 character length	212BYT	18
212 digital loopback	212BYT	18
212 enable	SQUBYT	7
212 mode	SQUBYT	7
212 remote digital loopback	212BYT	18
212 reset option	212BYT	18



**SWBYT
\$C080
-16256
READ**

7 DV FROM TOUCH TONE	6 BSR 60 HZ INPUT	5 CARRIER DETECT	4 AUDIO DETECT	3 SW4	2 SW3	1 SW2	0 SW1
----------------------------------	----------------------------	------------------------	----------------------	----------	----------	----------	----------

Bit

- 7 DV FROM TOUCHTONE
 - 1 valid TouchTone present
 - 0 no valid TouchTone present (CLR DV high)
- 6 BSR 60 HZ INPUT
 - continuous 1 no 60 Hz
 - alternating 60 Hz
- 5 CARRIER DETECT
 - 1 carrier present
 - 0 no carrier present
- 4 AUDIO DETECT*
 - 1 no audio present
 - 0 audio present
- 3 FIRMWARE SWITCH 4
 - 1 open (off)
 - 0 closed (on)
- 2 FIRMWARE SWITCH 3
 - 1 open (off)
 - 0 closed (on)
- 1 FIRMWARE SWITCH 2
 - 1 open (off)
 - 0 closed (on)
- 0 FIRMWARE SWITCH 1
 - 1 open (off)
 - 0 closed (on)

*Audio detect input changes with speech, dial tone, and busy signal. The input to the audio detector is the output of the receive filter.



DACBYT
\$C080
-16256
WRITE

7 MOST SIGNIFI- CANT BIT	6	5	4	3	2	1	0 LEAST SIGNIFI- CANT BIT
--------------------------------------	---	---	---	---	---	---	---------------------------------------

8 DAC bits

TONBYT
\$C081
-16255
READ

7 Not Used	6 Not Used	5 Not Used	4 IRQ 29.12 HZ STATUS	3 TOUCH TONE BIT D8	2 TOUCH TONE BIT D4	1 TOUCH TONE BIT D2	0 TOUCH TONE BIT D1
------------------	------------------	------------------	-----------------------------------	---------------------------------	---------------------------------	---------------------------------	---------------------------------

Bit

- 7 Not used
- 6 Not used
- 5 Not used
- 4 IRQ 29.12 HZ STATUS
 - 1 IRQ has not occurred
 - 0 IRQ has occurred

3-0 TOUCHTONE BITS

digit	D8	D4	D2	D1
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1



digit	D8	D4	D2	D1
8	1	0	0	0
9	1	0	0	1
0	1	0	1	0
*	1	0	1	1
#	1	1	0	0
A	1	1	0	1
B	1	1	1	0
C	1	1	1	1
D	0	0	0	0

TIMING: Within 40 ms of a valid received tone pair the data bits D8, D4, D2, and D1 will become valid. Seven microseconds after the data bits have become valid, DV will go high. The data bits will remain valid and DV will remain high as long as a valid tone pair is present at the receiver input. Within 40 ms of the removal of a valid tone pair from the input, the decoder will recognize a valid pause. DV goes low approximately 45 ms after the tone pair is removed. The data bits will be cleared (set to 0) 4.4 ms after DV is lowered. The DV strobe will be of at least the same duration as the incoming tone pair.

HANDSHAKE MODE: In this mode, the DV strobe is polled at least once every 40 ms to determine whether a new valid tone pair has been detected. If DV is high, indicating that a new valid tone pair has been detected, store the coded data present at the data bits of the receiver and clear DV by pulsing CLRDV high. With some systems operating in the handshake mode, it may be desirable to know when the valid tone has ended. Ordinarily this would be indicated by the falling edge of DV. However, in the handshake mode, DV is cleared by the monitoring system each time a new valid tone is detected and, therefore, cannot be used to determine when a valid tone pair has ended. The termination of a valid tone pair in this case may be observed by detecting the clearing of the data bits. Since, in hexadecimal format (the mode normally used with a handshake interface), the all zero state represents a commonly unused tone pair (D), the end of a valid tone pair may be detected by sensing the all zero state.

NOTE: DV is read as SWBYT bit 7. DV is cleared in SHBYT bit 0.



SQUBYT
\$C081
-16255
WRITE

7 212 ENABLE	6 Not Used	5 212 MODE	4 TRMC	3 TRMC	2 TRMC	1 SSQ	0 MSQ
--------------------	------------------	------------------	-----------	-----------	-----------	----------	----------

Bit

- 7 212 ENABLE*
 - 1 212 disable
 - 0 212 enable
- 6 Not used
- 5 212 MODE
 - 1 212 originate
 - 0 212 answer
- 4-2 TAPE RECORDER MOTOR CONTROL
 - 111 motor on
 - 000 motor off
- 1 SPEAKER SQUELCH
 - 1 not squelched
 - 0 squelched
- 0 MICROPHONE SQUELCH
 - 1 squelched
 - 0 not squelched

*212 ENABLE is set to 1 in 212 analog loopback test (see 212BYT).

NOTE: When changing ANS/ORIG mode, first write word with bit 5 in desired state and with bit 7 high; then wait 5-10 msec, and write word with same state for bit 5 but with bit 7 low. Bit 7 must be left enabled (low) whenever operating in 212.



SHBYT
\$C082
-16254
WRITE

7 Not Used	6 Not Used	5 Not Used	4 Not Used	3 Not Used	2 Not Used	1 SH	0 CLR DV
------------------	------------------	------------------	------------------	------------------	------------------	---------	-------------

Bit

- 7 Not used
- 6 Not used
- 5 Not used
- 4 Not used
- 3 Not used
- 2 Not used
- 1 SWITCH HOOK
 - 1 off hook (holding telephone line)
 - 0 on hook (not holding telephone line)
- 0 CLEAR DV (Reset TouchTone)
 - 1 clear DV to 0
 - 0 DV set



BSRBYT
\$C083
-16253
WRITE

7 Not Used	6 Not Used	5 Not Used	4 Not Used	3 Not Used	2 IRQ 29.12 HZ ENABLE	1 BSR 120 KHZ OUTPUT	0 Not Used
------------------	------------------	------------------	------------------	------------------	--------------------------------	-------------------------------	------------------

Bit

- 7 Not used
- 6 Not used
- 5 Not used
- 4 Not used
- 3 Not used
- 2 IRQ 29.12 HZ ENABLE
 - 1 IRQ disabled/reset*
 - 0 IRQ enabled
- 1 BSR 120 KHZ OUTPUT
 - 1 disabled
 - 0 enabled
- 0 Not used

*To clear IRQ, Bit 2 must be disabled, then enabled immediately.



RECBYT
\$C089
-16247
WRITE

7 Not Used	6 CDT	5 MC6	4 MC5	3 MC4	2 MC3	1 MC2	0 MC1
------------------	----------	----------	----------	----------	----------	----------	----------

Bit

- 7 Not used
- 6 CARRIER DETECT TIMING*
 - 1 normal carrier
 - 0 fast carrier
- 5-0 MODE CONTROL BITS

mode	value
103 ANS	\$64
103 ANS (mark carrier detect)	\$60
103 ORIG	\$65
103 ORIG (mark carrier detect)	\$61
202	\$5C
202 (mark carrier detect)	\$58
DEAF (TDD)	\$2D
ANS V.21	\$66
ANS V.21 (mark carrier detect)	\$62
ORIG V.21	\$67
ORIG V.21 (mark carrier detect)	\$63
900 Hz detect	\$6F
dial tone & busy signal †	\$7E

*Normal carrier

103, V.21: on delay—150ms

off delay— 50ms

202, V.23: on delay— 38ms

off delay— 13ms

Fast carrier (must be used in TDD mode)

All modes: on delay—9ms

off delay—9ms

†This mode does not detect dial tone or busy but simply puts the receive filter in the frequency range of dial tone, busy signals, and ringing. The audio detect bit (SWBYT bit 4) should be looked at to determine what is dial tone, busy, or speech.



**CON
\$C08A
-16246
WRITE**

UART CONTROL BYTE

7 Not Used	6 PE	5 P2	4 P1	3 SB2	2 SB1	1 DB2	0 DB1
------------------	---------	---------	---------	----------	----------	----------	----------

Bit

- 7 Not used
- 6 PARITY ENABLE
 - 1 enabled
 - 0 disabled
- 5-4 PARITY*
 - 11 mark
 - 10 odd
 - 01 space
 - 00 even
- 3-2 NUMBER OF STOP BITS
 - 11 —
 - 10 2
 - 01 1.5
 - 00 1
- 1-0 NUMBER OF DATA BITS
 - 11 8
 - 10 6
 - 01 7
 - 00 5

*The parity bit is additional to the number of data bits.



**INDAT
\$C08B
-16245
READ**

RECEIVE DATA FROM UART

7 MOST SIGNIFI- CANT BIT	6	5	4	3	2	1	0 LEAST SIGNIFI- CANT BIT
--------------------------------------	---	---	---	---	---	---	---------------------------------------

**SPDBYT
\$C08B
-16245
WRITE**

7 TBR4	6 TBR3	5 TBR2	4 TBR1	3 RBR4	2 RBR3	1 RBR2	0 RBR1
-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------

Bit

7-4 TRANSMIT BAUD RATE*

3-0 RECEIVE BAUD RATE*

*baud rate	bit time	actual baud rate	percent error	value
1200	.831 ms	1202.75	.2	\$0
600	1.66 ms	601.38	.2	\$1
300	3.33 ms	300.68	.2	\$2
150	6.65 ms	150.34	.2	\$3
110	9.15 ms	109.27	.6	\$5
75	13.3 ms	75.17	.2	\$6
50	20.0 ms	50.11	.2	\$7
45.5	22.0 ms	46.26	1.7	\$4

NOTE: Transmit and receive baud rates are independent.



COM
\$C08C
-16244
WRITE

UART COMMAND REGISTER

7 Not Used	6 Not Used	5 Not Used	4 Not Used	3 T2	2 T1	1 R2	0 R1
------------------	------------------	------------------	------------------	---------	---------	---------	---------

Bit

- 7 Not used
- 6 Not used
- 5 Not used
- 4 Not used
- 3-2 TRANSMITTER
 - 11 xmit off
 - 10 xmit on, enable IRQ
 - 01 xmit on, disable IRQ
 - 00 xmit break (space)
- 1-0 RECEIVER
 - 11 —
 - 10 receive on, enable IRQ
 - 01 receive on, disable IRQ
 - 00 receiver off



ACBYT
\$C08D
-16243
READ

7 Not Used	6 Not Used	5 Not Used	4 Not Used	3 CTS	2 212 CARRIER DETECT	1 Not Used	0 RING DETECT
------------------	------------------	------------------	------------------	----------	-------------------------------	------------------	---------------------

Bit

- 7 Not used
- 6 Not used
- 5 Not used
- 4 Not used
- 3 CLEAR TO SEND
printer handshake (busy)
follows input
- 2 212 CARRIER DETECT
 - 1 carrier present
 - 0 no carrier present
- 1 Not used
- 0 RING DETECT*
 - 1 ring present
 - 0 ring not present

*If IRQ enabled on ring, IRQ occurs on leading edge of ring (when RING goes high).



XMTBYT
\$C08D
-16243
WRITE

7 Not Used	6 Not Used	5 Not Used	4 IRQ RING ENABLE	3 XM4	2 XM3	1 XM2	0 XM1
------------------	------------------	------------------	----------------------------	----------	----------	----------	----------

Bit

- 7 Not used
 6 Not used
 5 Not used
 4 IRQ RING ENABLE*
 1 disabled/reset
 0 enabled
 3-0 TRANSMITTER MODE
- | function | value | space Hz | mark Hz |
|------------------------|-------|----------|-------------|
| 103 ANS | \$0 | 2025 | 2225 |
| 103 ORIG | \$1 | 1070 | 1270 |
| CCITT ANS | \$2 | 1850 | 1650 |
| CCITT ORIG | \$3 | 1180 | 980 |
| DEAF (TDD) | \$4 | 1800 | 1400 |
| 202 SOFT CAR | \$5 | 900 | |
| 202 U.S. or CCITT V.23 | \$6 | 2100 | 1300 |
| REV CHAN (VIEWDATA) | \$7 | 450 | 390 |
| disabled | >\$7 | | off (quiet) |

*To clear IRQ, write 1 then 0.

OUTDAT
\$C08E
-16242
WRITE

TRANSMIT DATA TO UART

7 MOST SIGNIFI- CANT BIT	6	5	4	3	2	1	0 LEAST SIGNIFI- CANT BIT
--------------------------------------	---	---	---	---	---	---	---------------------------------------



STATUS
\$C08F
-16241
READ

UART STATUS REGISTER

7 S8	6 S7	5 S6	4 S5	3 S4	2 S3	1 S2	0 S1
---------	---------	---------	---------	---------	---------	---------	---------

Bit

- 7 OVERRUN ERROR
 1 error
 0 ok
 6 FRAMING ERROR
 1 error
 0 ok
 5 PARITY ERROR
 1 error
 0 ok
 4 TX DATA REG EMPTY
 1 empty
 0 full
 3 RX DATA REG FULL
 1 full
 0 empty
 2 TRANSMIT IRQ
 1 transmit IRQ
 0 no transmit IRQ
 1 RECEIVE IRQ
 1 receive IRQ
 0 no receive IRQ
 0 RING IRQ
 1 no IRQ
 0 IRQ



**DEVBYT
\$C08F
-16241
WRITE**

INPUT/OUTPUT SELECTION REGISTER*

7 Not Used	6 Not Used	5 Not Used	4 Not Used	3 TXD2	2 TXD1	1 RXD2	0 RXD1
------------------	------------------	------------------	------------------	-----------	-----------	-----------	-----------

Bit

- 7 Not used
- 6 Not used
- 5 Not used
- 4 Not used
- 3-2 FROM UART TXD TO
 - 11 —
 - 10 add-on 212
 - 01 printer port
 - 00 main modem
- 1-0 UART RXD TO
 - 11 —
 - 10 add-on 212
 - 01 printer port
 - 00 main modem

*Register must be written to when entering 212 mode.



**212BYT
\$C080
-16256
WRITE**

7 Not Used	6 Not Used	5 AL	4 DL	3 RDL	2 CL2	1 CL1	0 RST
------------------	------------------	---------	---------	----------	----------	----------	----------

Bit

- 7 Not used
- 6 Not used
- 5 ANALOG LOOPBACK
 - 1 analog loopback
 - 0 normal
- 4 DIGITAL LOOPBACK
 - 1 digital loopback
 - 0 normal
- 3 REMOTE DIGITAL LOOPBACK
 - 1 remote digital loopback
 - 0 normal
- 2-1 212 CHARACTER LENGTH*
 - 10 10 bits
 - 01 9 bits
- 0 RESET
 - 1 normal
 - 0 reset option

*Length includes one start and one stop bit.

NOTE: Procedure for writing

- 1) Write with bits as desired and bit 0 low.
- 2) Delay at least 500 ms.
- 3) Write with bit 0 high, all other bits unchanged.



LISA 2.5

CATPACK

```

0800      2      TTL "CATPACK"
0800      3      ;A PACKAGE OF ROUTINES THAT WILL SERVE
0800      4      ;AS EXAMPLES OF HOW TO ACCESS THE
0800      5      ;APPLE-CAT MODEM THROUGH ASSEMBLY
0800      6      ;LANGUAGE. THESE ROUTINES ARE DESIGNED
0800      7      ;TO BE EXAMPLES, AND YOU MAY WANT TO
0800      8      ;MODIFY THEM TO MEET YOUR NEEDS.
0800      9      ;
0800     10      ;BY GREG SEITZ
0800     11      ;COPYRIGHT (C) 1981 NOVATION INC.
0800     12      ;
0800     13      ;REV 1.1    13-FEB-82
0800     14      ;ALL COMMERCIAL RIGHTS RESERVED
0800     15      ;
0800     16      ;NOTE: SOME OF THE STORAGE LOCATIONS
0800     17      ;USED IN THIS PROGRAM MAY CONFLICT WITH
0800     18      ;APPLE-SOFT OR INTEGER BASIC SO IF YOU
0800     19      ;WILL BE USING EITHER YOU MUST MAKE SURE
0800     20      ;THAT NO CONFLICTS ARISE...
0800     21      ;
0800     22      ;WARNING: MOST OF THESE ROUTINES WILL
0800     23      ;RETURN WITH THE REGISTERS WIPED OUT.
0800     24      ;SO IF YOU NEED TO SAVE THE VALUE
0800     25      ;OF A REGISTER YOU SHOULD DO SO BEFORE
0800     26      ;CALLING ANY OF THESE ROUTINES.
0800     27      ;
00E9     28      DLYL      EPZ #E9
00EA     29      DLYH      EPZ #EA
00E5     30      F1F       EPZ #E5
00E6     31      F1I       EPZ #E6
00E7     32      F2F       EPZ #E7
00E8     33      F2I       EPZ #E8
0080     34      DACBYT    EQU #C080
0095     35      CMDTAB    EPZ #95
005D     36      CMD       EPZ #5D
0097     37      CMDLOC    EPZ #97
03FE     38      IRD       EQU #3FE
00E1     39      FRAC1     EPZ #E1
00E0     40      INT1      EPZ #E0
00E4     41      FRAC2     EPZ #E4
00E3     42      INT2      EPZ #E3
0004     43      C3        EPZ #04
0011     44      COUNT     EPZ #11
001C     45      TEMPX     EPZ #1C
001D     46      TEMPY     EPZ #1D
001F     47      CARSTS     EPZ #1F
00F6     48      ZCTMP     EPZ #F6
0002     49      CROSS     EPZ #02
FC58     50      HOME      EQU #FC58
0033     51      PROMPT    EPZ #33
FD6A     52      GETLN     EQU #FD6A
0010     53      C7        EPZ #10
0200     54      KEYBUF     EQU #200
0350     55      DIGBUF     EQU #350
FCAB     56      WAIT      EQU #FCAB
030C     57      DIALSW     EQU #30C
02F1     58      SLOT      EQU #2F1
C010     59      KEYCLR    EQU #C010
0018     60      XMTRAM     EPZ #18

```



LISA 2.5

CATPACK

```

C000     61      KEYDAT    EQU #C000
C08C     62      COM       EQU #C08C
C08D     63      XMTBYT    EQU #C08D
C08A     64      CON       EQU #C08A
C083     65      BSRBYT    EQU #C083
C082     66      SHBYT     EQU #C082
C081     67      TONBYT    EQU #C081
C081     68      SOUBYT    EQU #C081
C080     69      SWBYT     EQU #C080
C089     70      RECBYT    EQU #C089
C08B     71      INDAT     EQU #C08B
C08E     72      OUTDAT    EQU #C08E
C08B     73      SPDBYT    EQU #C08B
C08F     74      STATUS    EQU #C08F
C08D     75      ACBYT     EQU #C08D
C08F     76      DEVBYT    EQU #C08F
FD6D     77      COUT      EQU #FD6D
FD60     78      COUT1     EQU #FD60
0800     79      ;
0800     80      ;
0800     81      ;THE FOLLOWING ROUTINE IS A TYPICAL
0800     82      ;EXAMPLE OF WHAT YOU WOULD DO UPON
0800     83      ;ENTRY TO A MODEM PROGRAM. FIRST WE
0800     84      ;GET THE SLOT THEN WE SAVE IT AND
0800     85      ;JUMP TO OUR HANG UP ROUTINE.
0800     86      SLOTINIT    LDY #520          ;SLOT 2
0800     87      STY SLOT
0800     88      JSR RESIRO
0800     89      JSR HANG1      ;HANG UP
0800     90      LDA #500      ;CLEAR CARRIER STATUS (RAM VER
SION)
080D     91      STA CARSTS   ;TO WHATEVER IT IS NOW.
080F     92      RTS
0810     93      ;
0810     94      ;
0810     95      ;THE FOLLOWING ROUTINE IS THE OUTPUT
0810     96      ;TO MODEM ROUTINE. SEND A CHARACTER TO
0810     97      ;IT THROUGH THE ACCUMULATOR. IN A
0810     98      ;ACTUAL PROGRAM IT WOULD ALSO BE A GOOD
0810     99      ;IDEA TO CHECK FOR CARRIER BEFORE
0810    100      ;JUMPING TO THIS ROUTINE.
0810    101      TTOUT      PHA          ;SAVE THE INPUT
0810    102      LDY SLOT      ;GET THE SLOT OF THE MODEM
0810    103      LDA STATUS,Y  ;SEE IF MODEM READY
0810    104      AND #00010000 ;TO SEND A CHARACTER OUT
0810    105      BEQ TTOUT1     ;LOOP UNTIL READY
0810    106      ;NOW WE HAVE THE GO-AHEAD FROM THE MODEM
0810    107      ;SO LETS GIVE IT WHAT IT WANTS...
0810    108      PLA          ;RECOVER THE ACCUM.
0810    109      STA OUTDAT,Y  ;SEND IT ON TO THE MODEM.
0810    110      RTS          ;RETURN.
0820    111      ;
0820    112      ;THE FOLLOWING ROUTINE IS THE INPUT
0820    113      ;FROM MODEM ROUTINE. IT WILL RETURN
0820    114      ;A CHARACTER THROUGH THE ACCUM. AGAIN.
0820    115      ;IT WOULD BE A GOOD IDEA IF YOU TESTED
0820    116      ;FOR CARRIER BEFORE CALLING THIS ROUTINE

```

20



LISA 2.5

CATPACK

```

0820          117 TTIN:
0820 AC F1 02 118      LDY SLOT          ;GET THE SLOT OF THE MODEM
0823 B9 8F C0 119      TTIN1 LDA STATUS,Y    ;CHECK TO SEE IF
0826 29 08      120      AND #%00001000    ;A CHAR WAITING TO BE READ.
0828 F0 F9      121      BEQ TTIN1          ;NO SO KEEP WAITING.
082A          122 ;NOW WE HAVE THE GO-AHEAD FROM THE MODEM, SO LETS GET IT.
082A A5 1B      123      LDA XMTAM          ;GET RAM VERSION OF XMITTER
082C 99 8D C0 124      STA XMTBYT,Y        ;CLEAR STATUS PORT
082F B9 8B C0 125      LDA INDAT,Y         ;GET THE CHARACTER
0832 60          126      RTS              ;RETURN WITH A CHAR!
0833          127 ;
0833          128 ;ROUTINE TO CHECK FOR CARRIER
0833          129 ;IF NO CARRIER THEN CHECK FOR
0833          130 ;RING...AND ATTEMPT TO HOOK UP.
0833          131 ;IF WE AREN'T HOOKED UP AFTER ALL
0833          132 ;THAT, THEN WE RETURN WITH THE
0833          133 ;CARRY BIT CLEAR OTHERWISE WE
0833          134 ;RETURN WITH THE CARRY BIT SET TO
0833          135 ;INDICATE THAT WE ARE CONNECTED
0833          136 ;THIS ROUTINE ALSO DISPLAYS PROMPTS
0833          137 ;AS TO WHAT IS GOING ON.
0833          138 ;IT WILL ALSO DETECT A CHANGE IN THE
0833          139 ;STATUS OF THE CARRIER AND HANGUP THE
0833          140 ;PHONE IF THE CARRIER WAS ON AND IS
0833          141 ;NOW OFF.
0833 20 FE 08    142      HANGUP JSR HANG1        ;HANG UP PHONE
0836 20 2B 09    143      JSR PRINTIT    ;TELL THE USER THAT THE FONE H
AS
0839 8D          144      HEX 8D          ;HUNG UP
083A C1 D0 D0    145      ASC "APPLE-CAT:HUNG UP"
083D CC C5 AD
0840 C3 C1 D4
0843 BA C8 D5
0846 CE C7 A0
0849 D5 D0
084B 8D 00      146      HEX 8D00
084D 4C 62 08    147      JMP CHKRNG          ;CONTINUE ON...
0850 38          148      SEC              ;SET CARRY TO INDICATE THAT
0851 60          149      RTS              ;WE HAVE A CARRIER AND RETURN
0852 AC F1 02    150      CARCHK LDY SLOT        ;GET THE CARD SLOT
0855 B9 80 C0    151      LDA SWBYT,Y        ;GET CARRIER BYTE
0858 29 20      152      AND #%00100000    ;MASK CARRIER BIT
085A D0 F4      153      BNE CARFND        ;WE HAVE A CARRIER!
085C A5 1F      154      LDA CARSTS        ;GET OLD STATUS
085E 29 20      155      AND #%00100000    ;SEE IF IT WAS ON
0860 D0 D1      156      BNE HANGUP        ;IT WAS ON SO WE HANG UP THE P
HONG
0862 B9 8D C0    157      CHKRNG LDA ACBYT,Y    ;NO CARRIER, SO CHECK FOR RING
0865 29 01      158      AND #%000000001    ;MASK RING BIT.
0867 F0 6E      159      BEQ NOCAR          ;PHONE NOT RINGING.
0869 20 2B 09    160      JSR PRINTIT
086C 8D          161      HEX 8D
086D 41 50 50    162      ASC 'APPLE-CAT:RING..
0870 4C 45 2D
0873 43 41 54
0876 3A 52 49
0879 4E 47 2E
087C 2E

```



LISA 2.5

CATPACK

```

087D 8D 00      163      HEX 8D00
087F B9 8D C0    164      FINRNG LDA ACBYT,Y    ;LET IT FINISH RINGING
0882 29 01      165      AND #%000000001    ;MASK RING BIT
0884 D0 F9      166      BNE FINRNG        ;STILL RINGING
0886          167 ;PHONE IS DONE RINGING, NOW WE WILL
0886          168 ;WAIT FOR A CARRIER TO COME IN.
0886 20 5F 09    169      JSR INIT          ;TURN ON MODEM
0889          170 ;
0889          171 ;GIVE OTHER SIDE 20 SECS TO GET
0889          172 ;CARRIER GOING.
0889 20 2B 09    173      JSR PRINTIT
088C 8D          174      HEX 8D
088D 41 50 50    175      ASC 'APPLE-CAT:AWAIT CARRIER'
0890 4C 45 2D
0893 43 41 54
0896 3A 41 57
0899 41 49 54
089C 20 43 41
089F 52 52 49
08A2 45 52
08A4 00          176      HEX 00
08A5 A2 2B      177      LDX #!40
08A7 AC F1 02    178      ANSLP1 LDY SLOT
08AA B9 80 C0    179      LDA SWBYT,Y    ;GET CARRIER BYTE.
08AD 29 20      180      AND #%00100000    ;MASK CARRIER BIT.
08AF D0 2C      181      BNE PRICAR        ;WE HAVE A CARRIER!
08B1 A9 05      182      LDA #!5        ;.5 SEC
08B3 20 19 09    183      JSR WAIT1        ;WAIT ROUTINE
08B6 CA          184      DEX
08B7 D0 EE      185      BNE ANSLP1    ;STILL HAVE TIME LEFT
08B9          186 ;NO CARRIER WAS DETECTED...
08B9 20 2B 09    187      JSR PRINTIT
08BC 8D 8D      188      HEX 8D8D
08BE 41 50 50    189      ASC 'APPLE-CAT:NO CARRIER'
08C1 4C 45 2D
08C4 43 41 54
08C7 3A 4E 4F
08CA 20 43 41
08CD 52 52 49
08D0 45 52
08D2 8D 00      190      HEX 8D00
08D4 20 33 08    191      JSR HANGUP        ;HANG UP FONE.
08D7 A9 00      192      NOCAR LDA #00        ;NO CARRIER
08D9 85 1F      193      STA CARSTS
08DB 18          194      CLC
08DE          ;CLEAR CARRY TO INDICATE NO CA
RRIE
08DC 60          195      RTS              ;RETURN TO CALLING ROUTINE
08DD 20 2B 09    196      PRICAR JSR PRINTIT
08E0 8D 8D      197      HEX 8D8D
08E2 41 50 50    198      ASC 'APPLE-CAT:CONNECTED'
08E5 4C 45 2D
08E8 43 41 54
08EB 3A 43 4F
08EE 4E 4E 45
08F1 43 54 45
08F4 44
08F5 8D 00      199      HEX 8D00

```




LISA 2.5

CATPACK

```

08F7 A9 20 200 LDA #2001000000 ;SET CARRIER STATUS TO ON
08F9 85 1F 201 STA CARSTS
08FB 4C 50 08 202 JMP CARFND ;ACKNOWLEDGE THAT WE HAVE A CA
RRIE
08FE AC F1 02 203 HANG1 LDY SLOT
0901 78 204 SEI ;TURN OFF INTERRUPTS
0902 A9 00 205 LDA #00 ;PUT THE PHONE ON THE HOOK
0904 99 82 C0 206 STA SHBYT,Y
0907 A9 01 207 LDA #2000000001 ;MIC & SPK
0909 99 81 C0 208 STA S0UBYT,Y ;OFF.
090C A9 0F 209 LDA #*0F ;MAKE SURE THAT WE DON'T
090E 99 83 C0 210 STA BSRBYT,Y ;GET ANY INTERRUPTS FROM
0911 A9 1F 211 LDA #*1F ;BSR & TURN OFF XMITTER
0913 99 8D C0 212 STA XMTBYT,Y
0916 85 1B 213 STA XMTIRAM
0918 60 214 RTS ;RETURN
0919 215 ;-----
0919 216 ;ENTER THIS ROUTINE WITH # OF 100MS
0919 217 ;TIME PERIODS IN THE ACCUM.
0919 86 1C 218 WAIT1 STX TEMPX ;SAVE X
091B 84 1D 219 STY TEMPY ;SAVE Y
091D A8 220 TAY ;MOVE COUNT TO Y REG
091E A9 C3 221 WAIT1A LDA #1195 ;DO 100MS WAIT
0920 20 A8 FC 222 JSR WAIT ;DELAY
0923 88 223 DEY
0924 D0 F8 224 BNE WAIT1A ;Y=0? NO-GO BACK
0926 A6 1C 225 LDX TEMPX ;RESTORE X
0928 A4 1D 226 LDY TEMPY ;RESTORE Y
092A 60 227 RTS ;RETURN
092B 228 ;-----
092B 229 ;ROUTINE TO PRINT OUT A STRING
092B 230 ;OF CHARS ENDING WITH A #00
092B 84 1D 231 PRINT1 STY TEMPY ;SAVE Y
092D 86 1C 232 STX TEMPX ;SAVE X
092F 68 233 PLA
0930 85 04 234 STA C3
0932 68 235 PLA
0933 85 05 236 STA C3+*1
0935 A0 01 237 LDY #1
0937 B1 04 238 LDA (C3),Y
0939 C9 00 239 CMP #00
093B F0 08 240 BEQ QUIT
093D 09 B0 241 ORA #*80 ;SET HIGH BIT
093F 20 F0 FD 242 JSR COUT1
0942 20 58 09 243 JSR UPDATE
0945 4C 37 09 244 JMP PRINT2
094B 20 58 09 245 QUIT JSR UPDATE
094B A5 05 246 LDA C3+*1
094D 48 247 PHA
094E A5 04 248 LDA C3
0950 48 249 PHA ;RESTORED THE STACK
0951 A6 1C 250 LDX TEMPX ;RESTORE X
0953 A4 1D 251 LDY TEMPY ;RESTORE Y
0955 A9 01 252 LDA #1 ;LEAVE WITH ACCUM=1 ALWAYS
0957 60 253 RTS
0958 E6 04 254 UPDATE INC C3
095A D0 02 255 BNE QUITR

```



LISA 2.5

CATPACK

```

095C E6 05 256 INC C3+*1
095E 60 257 QUITR RTS
095F 258 ;-----
095F 259 ;FOLLOWING IS A TYPICAL ROUTINE THAT
095F 260 ;YOU WOULD JUMP TO WHEN YOU WANTED TO
095F 261 ;INITIALIZE THE MODEM FOR DATA MODE.
095F 262 ;
095F 78 263 INIT SEI ;MAKE SURE NO INTERRUPTS OCCUR.
0960 AC F1 02 264 LDY SLOT ;GET SLOT NUMBER
0963 A9 01 265 LDA #2000000001 ;TURN OFF MIC & SPEAKER.
0965 99 81 C0 266 STA S0UBYT,Y
0968 A9 02 267 LDA #2000000010 ;TELL MODEM TO
096A 99 82 C0 268 STA SHBYT,Y ;PICK UP THE PHONE
096D A9 06 269 LDA #2000000110 ;DISABLE BSR & 30HZ INTERRUPTS.
096F 99 83 C0 270 STA BSRBYT,Y
0972 A9 64 271 LDA #*64 ;ANS 103
0974 99 89 C0 272 STA RECBYT,Y ;MODE FOR THE RECEIVER.
0977 A9 07 273 LDA #2000000111 ;8 DATA + 1.5 STOP
0979 99 8A C0 274 STA C0N,Y ;BITS, NO PARITY.
097C A9 22 275 LDA #*22 ;SET SPEED TO 300 BAUD
097E 99 8B C0 276 STA SPDBYT,Y ;XMIT & REC'D
0981 A9 05 277 LDA #2000000101 ;XMIT & REC'D WITH NO
0983 99 8C C0 278 STA C0M,Y ;INTERRUPTS
0986 A9 10 279 LDA #*10 ;ANS 103
0988 99 8D C0 280 STA XMTBYT,Y ;MODE FOR THE TRANSMITTER.
098B 85 1B 281 STA XMTIRAM ;RAM VERSION OF SAME.
098D A9 00 282 LDA #*00 ;DIRECT DATA TO AND
098F 99 8F C0 283 STA DEVBYT,Y ;FROM MAIN MODEM.
0992 60 284 RTS
0993 285 ;-----
0993 286 ;ROUTINE TO INITIALIZE MODEM FOR VOICE
0993 287 ;MODE.
0993 AC F1 02 288 INITV LDY SLOT ;GET MODEMS SLOT
0996 A9 03 289 LDA #2000000011 ;TURN ON SWITCH HOOK
0998 99 82 C0 290 STA SHBYT,Y ;AND ENABLE DV.
099B A9 02 291 LDA #2000000010 ;TURN ON SPK & MIC.
099D 99 81 C0 292 STA S0UBYT,Y
09A0 A9 1F 293 LDA #2000111111 ;TURN OFF XMITTER.
09A2 99 8D C0 294 STA XMTBYT,Y
09A5 85 1B 295 STA XMTIRAM
09A7 60 296 RTS
09A8 297 ;-----
09A8 298 ;ROUTINE TO READ A TOUCHTONE FROM THE
09A8 299 ;MODEM. YOU MUST HAVE TOUCH TONE OPTION
09A8 300 ;TO USE THIS ROUTINE, ALSO MAKE SURE
09A8 301 ;THE SWITCH HOOK IS ENABLED TO RECIEVE
09A8 302 ;TOUCH TONE. EG. SEND A #2000000011 TO
09A8 303 ; SHBYT,Y.
09A8 304 ;THE CHAR WILL BE RETURNED THROUGH
09A8 305 ;THE ACCUMULATOR.
09A8 B9 80 C0 306 RECTONE LDA SWBYT,Y ;IS THERE A TONE
09AB 29 80 307 AND #*80 ;TONE PRESENT?
09AD F0 F9 308 BEQ RECTONE ;LOOP TILL WE GET ONE
09AF B9 81 C0 309 LDA TONBYT,Y ;GET TONE DATA
09B2 29 0F 310 AND #2000001111 ;LOOK AT LO-NIBBLE ONLY.
09B4 AA 311 TAX ;TRANSFER TO THE X REG

```



LISA 2.5

CATPACK

```
09B5 BD C5 09 312 LDA TTABLE,X :GET THE CHAR FROM THE TABLE
09B8 48 313 PHA :SAVE THE CHAR FOR A MOMENT
09B9 A9 02 314 LDA #2 :CLEAR DV BY
09BB 99 82 C0 315 STA SHBYT,Y :STROBING CLRDV
09BE A9 03 316 LDA #3 :ON AND THEN
09C0 99 82 C0 317 STA SHBYT,Y :OFF AGAIN
09C3 68 318 PLA :RECOVER CHAR.
09C4 60 319 RTS
09C5 B0 B1 B2 320 TTABLE ASC "01234567890**ABCD"
09C8 B3 B4 B5
09CB B6 B7 B8
09CE B9 B0 AA
09D1 A3 C1 C2
09D4 C3 C4
09D6 321 :-----
09D6 322 :PRINTER OUTPUT ROUTINE: THIS ROUTINE
09D6 323 :ASSUMES THAT YOU HAVE ALREADY CORRECTLY
09D6 324 :SET UP THE PRINTER INTERFACE (SPEED,
09D6 325 :PARITY,BITS, ETC) BEFORE CALLING THIS
09D6 326 :ROUTINE.
09D6 327 :SEND THE CHAR TO BE XMITTED THROUGH
09D6 328 :THE ACCUMULATOR.
09D6 AC F1 02 329 PRINTOUT LDY SLOT
09D9 B9 8D C0 330 LDA ACBYT,Y :GET BYTE WITH HANDSHAKE BIT.
09DC 29 08 331 AND #%00001000 :MASK OUT BIT
09DE F0 F6 332 BEQ PRINTOUT :LOOP TILL READY.
09E0 20 10 08 333 JSR TTOUT :BRANCH TO THE CHAR OUT SUBROUTINE
TINE :RETURN
09E3 60 334 RTS
09E4 335 :-----
09E4 336 :FOLLOWING IS A COMPLETE DIALING
09E4 337 :PACKAGE, IT WILL HANDLE BOTH TOUCH
09E4 338 :TONE AS WELL AS PULSE DIALING,
09E4 339 :THIS ROUTINE IS FREE STANDING, SO
09E4 340 :THERE IS NO NEED TO SEND ANY CHARS
09E4 341 :TO IT, HOWEVER YOU MAY WANT TO
09E4 342 :BY-PASS THE MENU ROUTINES AND JUMP
09E4 343 :DIRECTLY INTO THE DIALING PORTION.
09E4 344 :
09E4 345 :THE DIALING ROUTINE WILL RETURN
09E4 346 :A #01 IN THE ACCUMULATOR IF THE
09E4 347 :DIALING WAS ABORTED, IF IT WAS
09E4 348 :COMPLETED IT WILL RETURN A #00
09E4 349 :NOTE: THIS ROUTINE WILL RETURN WITH
09E4 350 :INTERRUPTS OFF, SO YOU MAY NEED TO
09E4 351 :TURN THEM BACK ON AFTER CALLING THIS
09E4 352 :IF YOU ARE USING INTERRUPTS
09E4 353 :
09E4 8A 354 IROROUT TXA :SAVE X
09E5 48 355 PHA
09E6 98 356 TYA :SAVE Y
09E7 48 357 PHA :GET SLOT
09E8 AC F1 02 358 LDY SLOT :GET TONEBYTE
09EB B9 81 C0 359 LDA TONBYT,Y :GET TONEBYTE
09EE 29 10 360 AND #%00010000 :MASK OUT 30 HZ BIT
09F0 D0 17 361 BNE RESTOR :RETURN
09F2 A9 04 362 LDA #%00000100 :RESET AND RE-
```



LISA 2.5

CATPACK

```
09F4 99 83 C0 363 STA BSRBYT,Y :ENABLE BSR
09F7 A9 00 364 LDA #%00000000 :IRO'S
09F9 99 83 C0 365 STA BSRBYT,Y
09FC A5 11 366 LDA COUNT :GET COUNT
09FE F0 04 367 BEQ OFF :DONE
0A00 C6 11 368 DEC COUNT :DECREMENT TIMER
0A02 D0 05 369 BNE RESTOR :RETURN
0A04 A9 04 370 LDA #%00000100 :TURN OFF 30HZ
0A06 99 83 C0 371 STA BSRBYT,Y :IRO'S
0A09 68 372 RESTOR PLA :PULL Y
0A0A A8 373 TAY :RESTORE Y
0A0B 68 374 PLA :PULL X
0A0C AA 375 TAX :RESTORE X
0A0D A5 45 376 LDA #45 :RECOVER ACCUMULATOR
0A0F 40 377 RTI :RETURN FROM INTERRUPT
0A10 A9 E4 378 SETIRO LDA #IROROUT :GET LO BYTE LOCATION
0A12 BD FE 03 379 STA IRO :SAVE LOW IRO BYTE
0A15 A9 09 380 LDA /IROROUT :GET HI BYTE
0A17 BD FF 03 381 STA IRO+1 :SAVE HI BYTE
0A1A 60 382 RTS :GO BACK
0A1B 383 :-----
0A1B 384 : HERE IS THE ACTUAL DIALING ROUTINE
0A1B 385 : SO YOU WOULD BRANCH HERE TO DIAL A
0A1B 386 : NUMBER.
0A1B 387 :
0A1B 20 10 0A 388 DIALER JSR SETIRO :SET UP THE INTERRUPT ROUTINE
0A1E 20 00 08 389 JSR SLOTINIT :TO MAKE IT FREE STANDING
0A21 20 58 FC 390 JSR HOME :CLEAR SCREEN
0A24 20 2E 09 391 JSR PRINTIT :PRINT WHAT FOLLOWS
0A27 0D 0D 392 BYT #D,#D
0A29 BC C4 BE 393 BYT "<D> DIAL A NUMBER",#D
0A2C A0 A0 C4
0A2F C9 C1 CC
0A32 A0 C1 A0
0A35 CE D5 CD
0A38 C2 C5 D2
0A3B 0D
0A3C BC D2 BE 394 BYT "<R> RE-DIAL LAST NUMBER",#D
0A3F A0 A0 D2
0A42 C5 AD C4
0A45 C9 C1 CC
0A48 A0 CC C1
0A4B D3 D4 A0
0A4E CE D5 CD
0A51 C2 C5 D2
0A54 0D
0A55 BC C5 D3 395 BYT "<ESC> ABORT DIALING",#D,#D
0A58 C3 BE A0
0A5B C1 C2 CF
0A5E D2 D4 A0
0A61 C4 C9 C1
0A64 CC C9 CE
0A67 C7 0D 0D 396 BYT "CHOICE?",0
0A6A C3 CB CF
0A6D C9 C3 C5
0A70 BF 00
```




LISA 2.5

CATPACK

```
0A72 20 C9 0E 397 JSR COMND1 ;COMMAND PROCESSOR
0A75 44 398 BYT 'D'
0A76 86 0A 399 ADR DIAL
0A78 52 400 BYT 'R'
0A79 8C 0A 401 ADR REDIAL
0A7B 1B 402 BYT #1B
0A7C 82 0A 403 ADR ABORT
0A7E BB 404 BYT " " ;END OF DATA
0A7F 4C 1B 0A 405 JMP DIALER ;LOOP BACK
0A82 20 4C 0D 406 ABORT JSR RESIRD ;TURN OFF INTERRUPTS
0A85 60 407 RTS ;RETURN TO CALLER
0A86 20 C0 0A 408 DIAL JSR DIALSB ;DIAL SUBROUTINE
0A89 4C 8F 0A 409 JMP DIALOT ;JUMP OVER NEXT COMMAND
0A8C 20 CA 0B 410 REDIAL JSR REDLSB ;RE-DIAL SUBROUTINE
0A8F AC F1 02 411 DIALOT LDY SLOT ;GET SLOT
0A92 A9 21 412 LDA #21 ;PUT RECEIVER IN 103 ORIG
0A94 99 89 C0 413 STA RECBYT,Y
0A97 4C 82 0A 414 JMP ABORT
0A9A 415 ;-----
0A9A 416 ;ROUTINE TO GET TELEPHONE # IN
0A9A 417 ;DIGIT BUFFER (#350-#389)
0A9A A9 BA 418 GETNO LDA #BA ;: PROMPT
0A9C 85 33 419 STA PROMPT ;PUT IN MONITR LOCATION
0A9E 20 6A FD 420 JSR GETLN ;GET A LINE OF TEXT
0AA1 A9 00 421 LDA #0 ;ZERO THE
0AA3 85 10 422 STA C7 ;INDEX COUNTER
0AA5 A6 10 423 GETNO1 LDX C7 ;GET THE VALUE OF THE INDEX CO
UNT
0AA7 E0 38 424 CPX #38 ;IF TOO LONG THEN TRUNCATE
0AA9 F0 0F 425 BEQ QUIT3
0AAB BD 00 02 426 LDA KEYBUF,X ;GET THE CHAR IN THE BUFFER
0AAE C9 8D 427 CMP #8D ;IS IT C/R
0AB0 F0 08 428 BEQ QUIT3 ;YES RETURN TO CALLER
0AB2 9D 50 03 429 STA DIGBUF,X ;STORE IN DIGIT BUFFER
0AB5 E6 10 430 INC C7 ;INC COUNTER
0AB7 4C A5 0A 431 JMP GETNO1 ;LOOP FOR MORE
0ABA A9 8D 432 QUIT3 LDA #8D ;PUT RETURN IN LAST LOC
0ABC 9D 50 03 433 STA DIGBUF,X
0ABF 60 434 RTS ;RETURN
0AC0 435 ;-----
0AC0 20 58 FC 436 DIALSB JSR HOME ;CLEAR SCREEN
0AC3 20 2B 09 437 JSR PRINTIT
0AC6 BC B0 AD 438 BYT "<0-9,*,#> DIGITS TO DIAL",#D
0ACC AC A3 BE
0ACF A0 C4 C9
0AD2 C7 C9 D4
0AD5 D3 A0 D4
0ADB CF A0 C4
0ADB C9 C1 CC
0ADE 0D
0ADF BC D4 BE 439 BYT "<T> TOUCH TONE SELECT",#D
0AE2 A0 A0 A0
0AE5 A0 A0 A0
0AE8 A0 D4 CF
0AEB D5 C3 C8
0AEE A0 D4 CF
0AF1 CE C5 A0
```



LISA 2.5

CATPACK

```
0AF4 D3 C5 CC
0AF7 C5 C3 D4
0AFA 0D
0AFB BC D0 BE 440 BYT "<P> PULSE DIAL SELECT",#D
0AFE A0 A0 A0
0B01 A0 A0 A0
0B04 A0 D0 D5
0B07 CC D3 C5
0B0A A0 C4 C9
0B0D C1 CC A0
0B10 D3 C5 CC
0B13 C5 C3 D4
0B16 0D
0B17 BC AB BE 441 BYT "<+> PAUSE FOR DIALTONE",#D
0B1A A0 A0 A0
0B1D A0 A0 A0
0B20 A0 D0 C1
0B23 D5 D3 C5
0B26 A0 C6 CF
0B29 D2 A0 C4
0B2C C9 C1 CC
0B2F D4 CF CE
0B32 C5 0D
0B34 BC C0 BE 442 BYT "<@> PAUSE FOR 2 SECONDS",#D,#D
0B37 A0 A0 A0
0B3A A0 A0 A0
0B3D A0 D0 C1
0B40 D5 D3 C5
0B43 A0 C6 CF
0B46 D2 A0 B2
0B49 A0 D3 C5
0B4C C3 CF CE
0B4F C4 D3 0D
0B52 0D
0B53 CE CF D4 443 BYT "NOTE: <P> & <T> IF USED MUST BE THE",#D
0B56 C5 BA A0
0B59 BC D0 BE
0B5C A0 A6 A0
0B5F BC D4 BE
0B62 A0 C9 C6
0B65 A0 D5 D3
0B68 C5 C4 A0
0B6B CD D5 D3
0B6E D4 A0 C2
0B71 C5 A0 D4
0B74 C8 C5 0D
0B77 C6 C9 D2 444 BYT "FIRST LETTERS IN THE NUMBER",#D
0B7A D3 D4 A0
0B7D CC C5 D4
0B80 D4 C5 D2
0B83 D3 A0 C9
0B86 CE A0 D4
0B89 C8 C5 A0
0B8C CE D5 CD
0B8F C2 C5 D2
0B92 0D
```



LISA 2.5

CATPACK

```

0B93 CE D5 CD 445      BYT "NUMBER TO DIAL",0
0B96 C2 C5 D2
0B99 A0 D4 CF
0B9C A0 C4 C9
0B9F C1 CC 00
0BA2 20 9A 0A 446      JSR GETNO      ;GET #
0BA5 A2 00 447          LDX #0          ;INIT INDEX
0BA7 BD 50 03 448      LDA DIGBUF,X    ;GET CHAR
0BAA C9 8D 449          CMP #8D         ;IS IT <C/R>
0BAC D0 03 450          BNE REDLSB1     ;NO-SKIP NEXT LINE.
0BAE 4C D1 0C 451      JMP DLAB        ;YES-JUMP TO ABORT
0BB1 C9 D4 452      REDLSB1 CMP #1T
0BB3 D0 0A 453          BNE REDLSB2
0BB5 48 454          PHA
0BB6 A9 00 455          LDA #00
0BB8 8D 0C 03 456      STA DIALSW
0BBB 68 457          PLA
0BBC 4C CA 0B 458      JMP REDLSB
0BBF C9 D0 459      REDLSB2 CMP #1P
0BC1 D0 07 460          BNE REDLSB
0BC3 48 461          PHA
0BC4 A9 01 462          LDA #01
0BC6 8D 0C 03 463      STA DIALSW
0BC9 68 464          PLA
0BCA 20 4C 0D 465      REDLSB JSR RESIRO  ;TURN OFF ACIA
0BCD 20 58 FC 466      JSR HOME      ;CLEAR SCREEN
0BD0 20 2B 09 467      JSR PRINTIT
0BD3 20 20 20 468      BYT " <ESC> TO ABORT",#D,#D
0BD6 3C 45 53
0BD9 43 3E 20
0BDC 54 4F 20
0BDF 41 42 4F
0BE2 52 54 0D
0BE5 0D
0BE6 0D 0D 0D 469      BYT #D,#D,#D,#D,"DIALING:",0
0BE9 0D 44 49
0BEC 41 4C 49
0BEF 4E 47 3A
0BF2 00
0BF3 AC F1 02 470      LDY SLOT      ;GET CARD INDEX
0BF6 A9 03 471          LDA #00000011 ;TURN OFF
0BF8 99 81 C0 472      STA S0UBYT,Y    ; MICROPHONE
0BFB A9 7E 473          LDA #7E       ;SET RECEIVER
0BFD 99 89 C0 474      STA RECBYT,Y    ; TO DIAL-TONE DETECT MODE
0C00 A9 1F 475          LDA #1F       ;SET XMITTER
0C02 99 8D C0 476      STA XMTBYT,Y    ; TO OFF
0C05 85 1B 477          STA XMTRAM     ;STORE IN RAM VERS. OF XMTBYT.
0C07 478
0C07 A9 00 479          LDA #0         ;TURN
0C09 99 82 C0 480      STA SHBYT,Y    ; OFF SH
0C0C 481
0C0C A9 1E 482          LDA #30       ;TURN OFF SH FOR
0C0E 20 19 09 483      JSR WAIT1     ; 3 SEC.
0C11 A9 03 484          LDA #3       ;TURN ON
0C13 99 82 C0 485      STA SHBYT,Y    ; SH AGAIN

```



LISA 2.5

CATPACK

```

0C16 486      ;-----
0C16 487      ;DIAL TONE DETECTION ROUTINE
0C16 20 BA 0E 488      DIAL1A JSR RDKEY      ;LOOK AT KEYBOARD
0C19 90 07 489          BCC DIAL1B     ;NO CHAR SO SKIP TO DIAL1A
0C1B C9 1B 490          CMP #1B       ;IS CHAR <ESC>
0C1D D0 03 491          BNE DIAL1B     ;NO-JUMP NEXT LINE.
0C1F 4C D1 0C 492      JMP DLAB        ;YES-ABORT
0C22 B9 80 C0 493      DIAL1B LDA SWBYT,Y    ;GET BYTE WITH CARRIER
0C25 29 10 494          AND #00010000 ;MASK ZERO CROSSING BIT
0C27 D0 ED 495          BNE DIAL1A     ;IF NOT DIAL TONE THEN LOOP
0C29 496      ;OTHERWISE MAKE SURE DIAL TONE IS
0C29 497      ;AROUND FOR 1 SEC.
0C29 A9 1D 498          LDA #29
0C2B 20 21 0D 499      JSR COUNTM      ;SET UP
; 1 SEC. WAIT + COUNT 0-CROSSI
NGS.
0C2E C9 01 500          CMP #1
0C30 D0 E4 501          BNE DIAL1A     ;1 CROSSING MEANS DIAL TONE.
0C32 502      ;START OVER AGAIN
0C32 A2 00 503          LDX #0         ;INIT INDEX COUNTER.
0C34 BD 50 03 504      DIAL1  LDA DIGBUF,X    ;GET 1'ST CHAR.
0C37 C9 8D 505          CMP #8D         ;IS IT <C/R>
0C39 F0 70 506          BEQ DIAL0V     ;YES-GO TO END ROUTINE.
0C3B 48 507          PHA              ;SAVE CHAR
0C3C 20 ED FD 508      DIAL2  JSR COUNT      ;PRINT CHAR
0C3F 68 509          PLA              ;RECOVER CHAR
0C40 E8 510          INX              ;INCREMENT CHAR. COUNTER
0C41 48 511          PHA              ;SAVE A IN STACK
0C42 20 BA 0E 512      JSR RDKEY      ;LOOK AT KEYBOARD
0C45 90 07 513          BCC DIAL3     ;NO CHAR SO SKIP TO DIAL3
0C47 C9 1B 514          CMP #1B       ;IS CHAR <ESC>
0C49 D0 03 515          BNE DIAL3     ;NO-SKIP NEXT CODE.
0C4B 4C D0 0C 516      DIAL3  JMP DIALAB     ;YES-TO ABORT MESSAGE.
0C4E 68 517          PLA              ;RESTORE A
0C4F C9 C0 518          CMP #0C       ;IS IT A '0'
0C51 D0 07 519          BNE PLUS      ;NO-SO SKIP TO PLUS
0C53 48 520          PHA              ;SAVE CHAR IN STACK
0C54 A9 14 521          LDA #20       ;WAIT FOR
0C56 20 19 09 522      JSR WAIT1     ; 2 SEC.
0C59 68 523          PLA              ;RECOVER CHAR
0C5A C9 AB 524      PLUS    CMP #AB      ;IS IT '+' (PAUSE)
0C5C D0 1B 525          BNE DIAL3A     ;NO-GOTO DIAL3A
0C5E 526      ;-----
0C5E 527      ;DIAL TONE DETECTION ROUTINE FOR ' + '
0C5E 48 528          PHA              ;SAVE CHAR IN STACK
0C5F 20 BA 0E 529      DTD1  JSR RDKEY      ;LOOK AT KEYBOARD
0C62 90 04 530          BCC DTD2     ;NO CHAR SO SKIP TO DIAL1A
0C64 C9 1B 531          CMP #1B       ;IS CHAR <ESC>
0C66 F0 68 532          BEQ DIALAB     ;YES-ABORT
0C68 B9 80 C0 533      DTD2  LDA SWBYT,Y    ;GET BYTE WITH CARRIER
0C6B 29 10 534          AND #00010000 ;MASK ZERO CROSSING BIT
0C6D D0 F0 535          BNE DTD1     ;IF NOT DIAL TONE THEN LOOP
0C6F 536      ;OTHERWISE MAKE SURE DIAL TONE IS
0C6F 537      ;AROUND FOR 1 SEC.
0C6F A9 1D 538          LDA #29
0C71 20 21 0D 539      JSR COUNTM      ;SET UP
; 1 SEC. WAIT + COUNT 0-CROSSI
NGS.
0C74 C9 01 540          CMP #1
0C76 D0 E7 541          BNE DTD1     ;1 CROSSING MEANS DIAL TONE.
;START OVER AGAIN

```




LISA 2.5

CATPACK

```

0C7B 68      542      PLA      ;RECOVER CHAR
0C7C 69      543      ;
0C7D C9 AA    544      DIAL3A  CMP #*AA      ;IS IT '*' ?
0C7E D0 04    545      DNE D3A1      ;NO-SKIP AHEAD
0C7F A9 0A    546      LDA #*A      ;YES-PUT #A IN ACCUM
0C80 D0 1B    547      BNE DIAL3B      ; AND SEND IT
0C81 C9 A3    548      D3A1  CMP #*A3      ;IS IT '*' ?
0C83 D0 04    549      DNE D3A2      ;NO-SKIP AHEAD
0C85 A9 0B    550      LDA #*B      ;YES-PUT #B IN ACCUM
0C87 D0 10    551      BNE DIAL3B      ; AND SEND IT
0C89          552      ;
0C8F 3B      553      D3A2  SEC      ;PREPARE FOR SUBTRACT WITHOUT
CARR          554      SBC #*B0
0C8A E9 D0    555      BMI DIAL5
0C8C 30 1A    556      CMP #*A
0C8E C9 0A    557      BPL DIAL5      ;GO GET NEXT # (TOO LARGE)
0C90 10 16    558      PHA          ;SAVE # ON STACK
0C92 4B      559      LDA DIALSW      ;IS TOUCH-TONE ENABLED(=0)
0C94 AD 0C 03 560      BNE DIAL4      ;NO-GO TO DIAL4
0C96 D0 07    561      PLA          ;YES-GET A
0C98 6B      562      DIAL3B JSR TONE      ;SEND THE TOUCH-TONE
0C9A 20 40 0E 563      JMP DIAL1      ;GO GET NEXT CHAR
0C9C 4C 34 0C 564      DIAL4  LDA #B      ;BETWEEN-DIGIT
0C9E A9 0B    565      JSR WAIT1      ;WAIT OF 800 MS
0CA1 20 19 09 566      PLA          ;RECOVER A REG
0CA3 6B      567      JSR PULSIT      ;GO DIAL #
0CA5 20 F4 0C 568      DIAL5  JMP DIAL1      ;GO GET NEXT #
0CA7 4C 34 0C 569      ;
0CAB 20 2B 09 570      DIALOV JSR PRINTIT ;DIALING-OVER ENDING
0CAE 0D 0D 44 571      BYT #D,#D,'DIALING COMPLETED',0
0CB1 49 41 4C
0CB4 49 4E 47
0CB7 20 43 4F
0CBA 4D 50 4C
0CBD 45 54 45
0CC0 44 00
0CC2 AC F1 02 572      LDY SLOT      ;GET CARD INDEX
0CC5 A9 02    573      LDA #2000000010 ;TURN ON MIC
0CC7 99 81 C0 574      STA S0UBYT,Y ; AND SPKR
0CCA 20 4C 0D 575      JSR RESIRO ;ASSURE INTERRUPTS OFF
0CCD A9 00    576      LDA #0 ;SIGNAL DIALING COMPLETED
0CCF 60      577      RTS ;RETURN FROM DIALING ROUTINE.
0CD0 6B      578      DIALAB PLA ;RECOVER CHAR
0CD1 AC F1 02 579      DLAB  LDY SLOT ;GET CARD INDEX
0CD4 A9 00    580      LDA #0 ;TURN OFF
0CD6 99 82 C0 581      STA SHBYT,Y ; SWITCH HOOK
0CD9 A9 01    582      LDA #2000000001 ;TURN OFF MIC
0CDB 99 81 C0 583      STA S0UBYT,Y ; AND SPKR.
0CDE 20 2B 09 584      JSR PRINTIT
0CE1 0D 0D 44 585      BYT #D,#D,'DIALING ABORTED',0
0CE4 49 41 4C
0CE7 49 4E 47
0CEA 20 41 42
0CF0 4F 52 54
0CFD 45 44 00
0CF3 60      586      RTS ;RETURN FROM DIALING ROUTINE.

```



LISA 2.5

CATPACK

```

0CF4          587      ;ACCU. HAS #01 TO INDICATE ABORT.
0CF4          588      ;
0CF4 86 1C    589      PULSIT STX TEMPX ;SAVE X AND
0CF6 84 1D    590      STY TEMPY ;Y REGS
0CF8 C9 00    591      CMP #0 ;IS # 0 ?
0CFA D0 02    592      BNE PULSE1 ;NO-SKIP NEXT STEP
0CFC A9 0A    593      LDA #10 ;YES-MAKE IT 10
0CFE AA      594      PULSE1 TAX ;TRANSFER A TO X
0CFF AC F1 02 595      PULSE2 LDY SLOT ;GET CARD INDEX
0D02 A9 00    596      LDA #0 ;TURN
0D04 99 82 C0 597      STA SHBYT,Y ;OFF SH
0D07 A9 96    598      LDA #150 ;FOR
0D09 20 A8 FC 599      JSR WAIT ;60 MS.
0D0C A9 03    600      LDA #3 ;TURN ON
0D0E 99 82 C0 601      STA SHBYT,Y ;SH AGAIN
0D11 CA      602      DEX ;DECREMENT X
0D12 F0 0B    603      BEQ PULSE3 ;IF X=0 THEN WE ARE DONE
0D14 A9 7B    604      LDA #123 ;WAIT FOR
0D16 20 A8 FC 605      JSR WAIT ;40 MS.
0D19 4C FF 0C 606      JMP PULSE2 ;DO IT AGAIN
0D1C A6 1C    607      PULSE3 LDX TEMPX ;RESTORE X AND
0D1E A4 1D    608      LDY TEMPY ;Y REGS.
0D20 60      609      RTS ;RETURN
0D21          610      ;
0D21          611      COUNTM:
0D21          612      ;IN THE ACCUMULATOR.
0D21          613      ;THE # OF ZERO CROSSINGS IS RETURNED IN THE ACCUM.
0D21 85 11    614      STA COUNT ;PUT ACCUM IN COUNT REGISTER
0D23 9B      615      TYA ;SAVE Y-REG
0D24 4B      616      PHA ; IN STACK
0D25 AC F1 02 617      LDY SLOT ;GET CARD INDEX
0D28 A9 00    618      LDA #0 ;TURN ON
0D2A 85 02    619      STA CROSS ;ZERO TRANSITION COUNTER
0D2C 99 83 C0 620      STA BSRBYT,Y ; 30 HZ IRQ'S
0D2F A9 10    621      LDA #2000010000 ;INITIALIZE
0D31 85 F6    622      STA ZCTMP ;ZERO TEMP LOCATION TO 1.
0D33 5B      623      CLI ;ENABLE IRQS
0D34 A5 11    624      COUNT1 LDA COUNT ;IS IRQ DELAY ROUTINE DONE?
0D36 F0 0F    625      BEQ COUNT3 ;YES-EXIT ROUTINE
0D38 B9 80 C0 626      COUNT2 LDA SWBYT,Y ;GET SWITCH BYTE
0D3B 29 10    627      AND #2000010000 ;MASK ZERO CROSSING BIT
0D3D C5 F6    628      CMP ZCTMP ;IS POLARITY SAME AS BEFORE.
0D3F F0 F3    629      BEQ COUNT1 ;YES-GO LOOP
0D41 85 F6    630      STA ZCTMP ;STORE NEW VALUE OF POLARITY
0D43 E6 02    631      INC CROSS ;NO-ADD 1 TO CROSSINGS.
0D45 D0 ED    632      BNE COUNT1 ;UNCONDIT. JMP TO COUNT1
0D47 6B      633      COUNT3 PLA ;RECOVER Y-REG
0D48 A8      634      TAY ; FROM STACK
0D49 A5 02    635      LDA CROSS ;LOAD ACCUM. WITH # OF 0-CROSS
ING5          636      RTS ;RETURN TO CALLER
0D4B 60      637      ;
0D4C          638      RESIRO SEI ;TURN OFF INTERRUPTS
0D4C 7B      639      LDY SLOT
0D50 A9 04    640      LDA #*04 ;TURN OFF
0D52 99 8C C0 641      STA COM,Y ;UART IRQ'S
0D55 60      642      RTS

```



LISA 2.5

CATPACK

```

0D56      643 ;NOTE:IT IS VERY IMPORTANT THAT THE
0D56      644 ;FOLLOWING ROUTINE BE LOCATED IN
0D56      645 ;A FULL PAGE (256 BYTES NO BOUNDARY
0D56      646 ;CROSSING) BECAUSE OF TIMING REQUIREMENTS
0E00      647 ORG $FF00:$100+* ;SKIP TO NEXT EVEN PAGE BOUNDR
Y
0E00      648 :
0E00      649 : *** TOUCH TONE ROUTINE ***
0E00      650 : MUST ALL BE IN ONE PAGE OF MEMORY
0E00      651 :
0E00      652 : 16-BYTE SINE LOOK-UP TABLE
0E00 40 58 653 SINE BYT $40,$58
0E02 6C 7A 654 BYT $6C,$7A
0E04 7F 7A 655 BYT $7F,$7A
0E06 6C 58 656 BYT $6C,$58
0E08 40 28 657 BYT $40,$28
0E0A 13 05 658 BYT $13,$05
0E0C 00 05 659 BYT $00,$05
0E0E 13 28 660 BYT $13,$28
0E10      661 :
0E10      662 : TOUCH-TONE CONSTANTS.
0E10      663 : (NEED TO BE ALTERED IF
0E10      664 : CPU CLOCK ISN'T 1.023 MHZ)
0E10      665 ;RE-CALCULATED FOR 93 MACHINE CYCLES
0E10      666 : 11/11/90
0E10      667 :
0E10 5E 01 668 TABL BYT $5E,$01
0E12 F1 01 669 BYT $F1,$01
0E14 04 01 670 D1 BYT $04,$01
0E16 C2 01 671 BYT $C2,$01
0E18 04 01 672 D2 BYT $04,$01
0E1A F1 01 673 BYT $F1,$01
0E1C 04 01 674 D3 BYT $04,$01
0E1E 26 02 675 BYT $26,$02
0E20 1F 01 676 D4 BYT $1F,$01
0E22 C2 01 677 BYT $C2,$01
0E24 1F 01 678 D5 BYT $1F,$01
0E26 F1 01 679 BYT $F1,$01
0E28 1F 01 680 D6 BYT $1F,$01
0E2A 26 02 681 BYT $26,$02
0E2C 3D 01 682 D7 BYT $3D,$01
0E2E C2 01 683 BYT $C2,$01
0E30 3D 01 684 D8 BYT $3D,$01
0E32 F1 01 685 BYT $F1,$01
0E34 3D 01 686 D9 BYT $3D,$01
0E36 26 02 687 BYT $26,$02
0E38 5E 01 688 D10 BYT $5E,$01
0E3A C2 01 689 BYT $C2,$01 ;#
0E3C 5E 01 690 D11 BYT $5E,$01
0E3E 26 02 691 BYT $26,$02 ;*
0E40      692 :
0E40      693 : TOUCHTONE GEN.
0E40      694 :
0E40 86 1C 695 TONE STX TEMPX ;SAVE X REG
0E42 84 1D 696 STY TEMPY ;SAVE Y REG
0E44 A2 04 697 LDX #4
0E46 86 EA 698 STX DLYH ;SET-UP INTERDIGIG DELAY

```



LISA 2.5

CATPACK

```

0E48 0A      699 MULT4 ASL ;MULTIPLY
0E49 0A      700 ASL ;DIGIT BY 4
0E4A AB      701 TAY ;PUT OFFSET IN Y
0E4B B9 10 0E 702 LDA TABL,Y ;GET 1ST FRACTION
0E4E 85 E5 703 STA F1F
0E50 CB      704 INY
0E51 B9 10 0E 705 LDA TABL,Y ;GET 2ND INTEGER
0E54 85 E6 706 STA F1I
0E56 CB      707 INY
0E57 B9 10 0E 708 LDA TABL,Y ;GET 2ND FRACTION
0E5A 85 E7 709 STA F2F
0E5C CB      710 INY
0E5D B9 10 0E 711 LDA TABL,Y ;GET 2ND INTEGER
0E60 85 E8 712 STA F2I
0E62 EA      713 NEXT NOP
0E63 EA      714 NOP ;PAD
0E64 EA      715 NOP ;BRANCH
0E65 EA      716 NOP ;ERROR
0E66 A5 E5 717 NEXT1 LDA F1F
0E68 18      718 CLC
0E69 65 E1 719 ADC FRAC1
0E6B 85 E1 720 STA FRAC1
0E6D A5 E6 721 LDA F1I
0E6F 65 E0 722 ADC INT1
0E71 85 E0 723 STA INT1
0E73 29 0F 724 AND #$F
0E75 AA      725 TAX
0E76 BD 00 0E 726 LDA SINE,X
0E79 48      727 PHA
0E7A A5 E7 728 LDA F2F
0E7C 18      729 CLC
0E7D 65 E4 730 ADC FRAC2
0E7F 85 E4 731 STA FRAC2
0E81 A5 E8 732 LDA F2I
0E83 65 E3 733 ADC INT2
0E85 85 E3 734 STA INT2
0E87 29 0F 735 AND #$F
0E89 AA      736 TAX
0E8A 68      737 PLA
0E8B 18      738 CLC
0E8C 7D 00 0E 739 ADC SINE,X
0E8F 29 F0 740 AND #$F0
0E91 09 04 741 ORA #$04
0E93 AC F1 02 742 LDY SLOT ;GET SLOT INDEX
0E96 99 80 C0 743 STA DACBYT,Y ;STORE TO DAC
0E99 C6 E9 744 DEC DLYL
0E9B D0 C5 745 BNE NEXT
0E9D C6 EA 746 DEC DLYH
0E9F D0 C5 747 BNE NEXT1
0EA1 A9 74 748 LDA #$74
0EA3 AC F1 02 749 LDY SLOT ;GET SLOT INDEX
0EA6 99 80 C0 750 STA DACBYT,Y ;STORE TO DAC
0EA9 A9 10 751 LDA #$10
0EAB 85 EA 752 DELAY2 STA DLYH ;INTERDIGIT
0EAD C6 E9 753 DLY1 DEC DLYL ;DELAY
0EAF D0 FC 754 BNE DLY1 ;ROUTINE

```

34



LISA 2.5

CATPACK

```

OEB1 C6 EA 755 DEC DLYH
OEB3 D0 F8 756 BNE DLY1
OEB5 A6 1C 757 LDX TEMPX ;RESTORE X
OEB7 A4 1D 758 LDY TEMPY ;AND Y REGS
OEB9 60 759 RTS
OEB A 760 ;-----
OEB A 761 ;THIS SECTION GETS A CHAR FROM THE KEYBD
OEB A 762 ;AND PRINTS BOTTOM LINES IF NECESSARY.
OEB A AD 00 C0 763 RDKEY LDA KEYDAT ;READ KEYBOARD
OEBD 30 02 764 BMI RDKEY1 ;YES. THEN GO TO RDKEY1
OEBF 18 765 CLC ;NO-SEND ROUTINE BACK
OECO 60 766 RTS ;WITH CARRY CLEAR TO MEAN NO C
HARACTOR
OEC1 AD 10 C0 767 RDKEY1 LDA KEYCLR ;CLEAR KEYBOARD STROBE AND STR
IF MSB
OEC4 AD 00 C0 768 LDA KEYDAT ;GET CHARACTER FROM KEYBOARD
OEC7 38 769 SEC ;AND RETURN TO ROUTINE
OEC8 60 770 RTS ;WITH CARRY SET TO MEAN GOT CH
AR.
OEC? 771 ;-----
OEC9 68 772 COMND1 PLA ;GET RETURN ADDRESS SO WE
OECA 85 95 773 STA CMDTAB ;KNOW WHERE DATA STARTS
OECB 68 774 PLA ;THEN STORE LOCATION IN
OECD 85 96 775 STA CMDTAB+1 ;THIS PAGE ZERO LOCATION
OECF AD 10 C0 776 LDA KEYCLR ;CLEAR KEYBOARD STROBE
OED2 20 BA OE 777 MENU2 JSR RDKEY ;READ KEYBOARD
OED5 90 FB 778 BCC MENU2 ;IF CARRY CLEAR, DO IT AGAIN
OED7 85 5D 779 STA CMD ;STORE IN ITS OWN LOCATION
OED9 A0 01 780 LDY #1 ;PUT 1 INTO Y
OEDB B1 95 781 CMD1 LDA (CMDTAB),Y ;GET CHAR INDEXED BY Y
OEDD C9 3B 782 CMP #' ; IS IT THE END OF COMMANDS
OEDF F0 21 783 BEQ CMD4 ;YES-THEN END
OEE1 C5 5D 784 CMP CMD ;IS IT A COMMAND
OEE3 F0 0C 785 BEQ CMD2 ;YES-GO OFF AND DO IT
OEE5 20 0C 0F 786 JSR UPDAT1 ;INCREMENT
OEE8 20 0C 0F 787 JSR UPDAT1 ;CMDTAB LOCATION
OEEB 20 0C 0F 788 JSR UPDAT1 ;THREE TIMES
OEEE 4C DB OE 789 JMP CMD1
OEF1 20 0C 0F 790 CMD2 JSR UPDAT1
OEF4 B1 95 791 LDA (CMDTAB),Y
OEF6 85 97 792 STA CMDLOC
OEF8 20 0C 0F 793 JSR UPDAT1
OEFB B1 95 794 LDA (CMDTAB),Y
OEFD 85 98 795 STA CMDLOC+1
OEFF 6C 97 00 796 JMP (CMDLOC)
OF02 20 0C 0F 797 CMD4 JSR UPDAT1 ;INCREMENT Y ONCE MORE
OF05 A5 96 798 LDA CMDTAB+1
OF07 48 799 PHA
OF0B A5 95 800 LDA CMDTAB
OF0A 48 801 PHA
OF0B 60 802 RTS
OF0C E6 95 803 UPDAT1 INC CMDTAB
OF0E D0 02 804 BNE UPDAT2
OF10 E6 96 805 INC CMDTAB+1
OF12 60 806 UPDAT2 RTS
OF13 807 END

```

***** END OF ASSEMBLY

